

577214

MARS SURVEYOR LANDING SITES "COLLABORATORY"

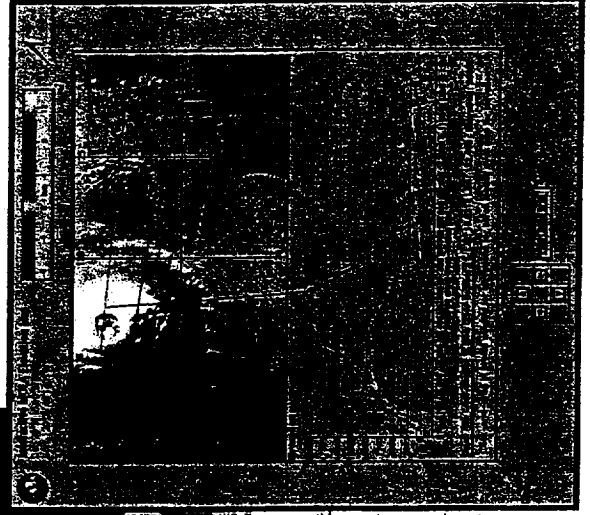
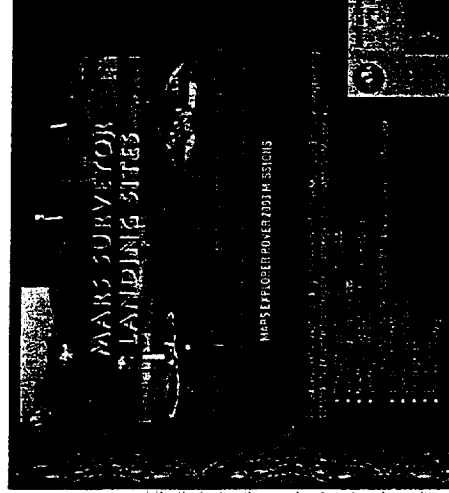


Ames Research Center | www.arc.nasa.gov

"Marsoweb" is a web-based facility for Mars researchers to collaborate on landing site selection studies for upcoming Mars Surveyor missions. Currently, Marsoweb serves as a comprehensive archive of data pertinent to selection of landing sites for the 2003 Mars Explorer Rover missions (MER 2003).

This interactive archive allows visual navigation of the candidate landing sites and various maps of data from the current Mars Global Surveyor (MGS) mission and from the Viking missions. Numerical querying is supported for much of this data. The MER 2003 archive also includes 3D VRML scenes in which the data has been texture-mapped onto terrain grids, as well as a repository of high-resolution images of the potential landing sites from the MGS Mars Orbiter Camera; this image repository includes Java-based image processing.

In addition to landing site selection resources, Marsoweb also features interactive archives of global data, which enable users to interrogate MGS data maps and geology maps, including the ability to query profiles created from user-drawn cross-sections in the maps. Near-term plans include implementing remote collaboration features and inclusion of Venus topography data.



Glenn Deardorff

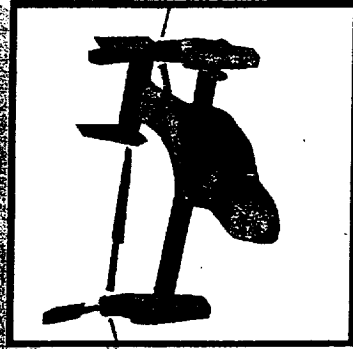
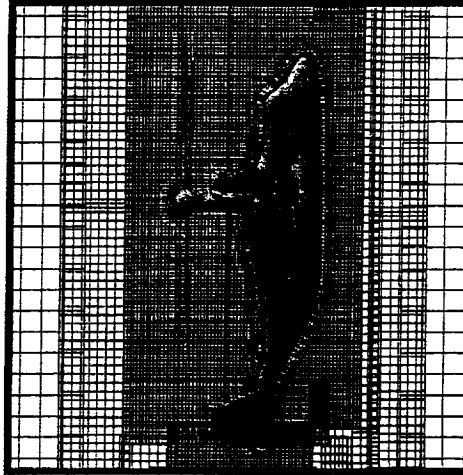
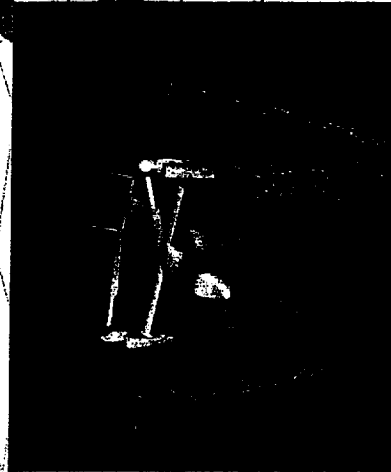
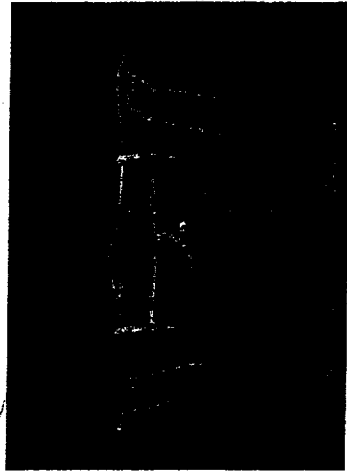


PARALLEL AND DISTRIBUTED CFD FOR UNSTEADY FLOWS WITH MOVING OVERSET GRIDS

Ames Research Center | www.arc.nasa.gov

This demonstration presents the development of a parallel and distributed computing (PDC) tool for large-scale unsteady moving body applications using overset structured grids, where the flow domain is decomposed into a union of multiblock meshes that may be in relative motion to one another.

Simulations of such moving grid systems demand executing an extensive computing procedure to update the overlap boundary data across the grid zones at every time step prior to flow computations. This is in contrast to simulations with static grids that only require one such calculation for all subsequent time iterations. An approach based on the MPI message-passing Chimera paradigm has been implemented in a NASA CFD code, called OVERFLOW-D, along with the Globus-MPICH-G toolkit that extends the execution environment to PDC. We are simulating a large-scale Navier-Stokes problem consisting of a moving body grid system of about 37 million points. A performance assessment indicates the feasibility of wide area networks like the Information Power Grid for this class of applications.



M. Tahed D. Gendari

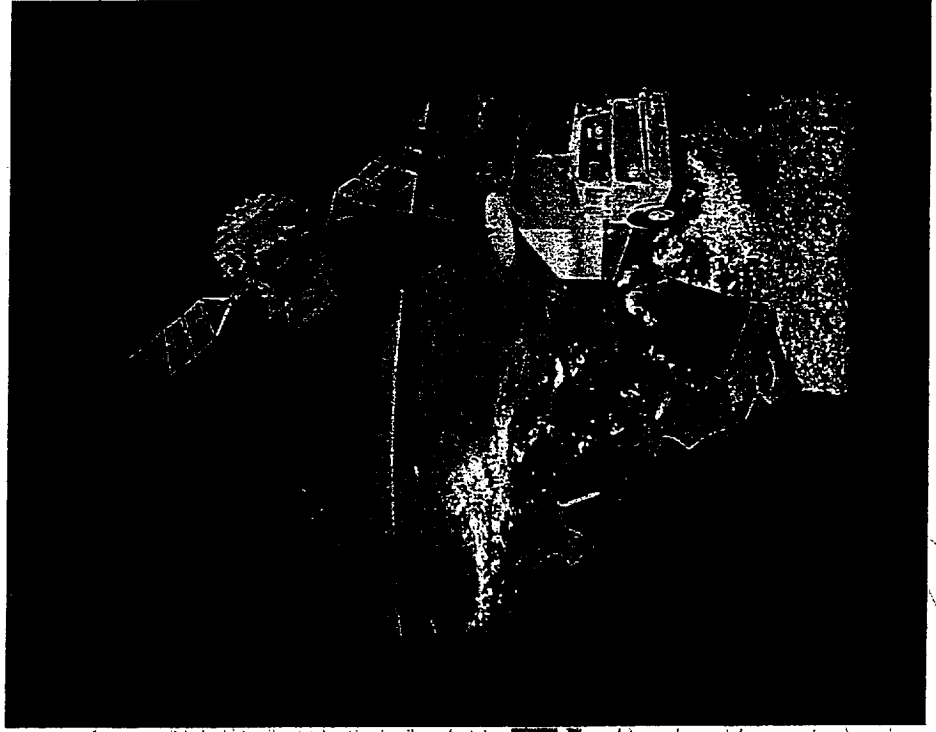


IP MULTICAST FOR SEAMLESS SUPPORT OF REMOTE SCIENCE

Ames Research Center | www.arc.nasa.gov

Scientific investigations at remote locations on Earth can be tremendously enhanced by connectivity to existing networks using standard protocols. Ready access to high-quality imagery and data improves the performance of the remote teams and increases the quality of their results. This demonstration shows the capability to both send and receive high-fidelity images using IP multicast (via geosynchronous satellites) with the NASA Research and Education Network/Glenn Research Center (NREN/GRC) Transportable Earth Station (TES). With IP multicast, the satellite and network capacities are used optimally, regardless of the number of participants, yet the distribution of the signal can be limited to a specific group.

In this demo, previously stored video content is sent via IP multicast to client systems. The receiving clients are commercial off-the-shelf systems. The 20 Mbps multicast can be sent via satellite from a server at GRC, and from a server that is located at the SC2001 exhibit hall. The TES has the capability of supporting two such streams (40 Mbps total) in each direction. The capability is applicable to scientific investigation field sites, experimental equipment evaluation tests, and supports temporary collaboration with a location that has limited high-speed communications capabilities.



Ken Freeman

CONSOLIDATED SUPERCOMPUTING MANAGEMENT OFFICE (COSMO)

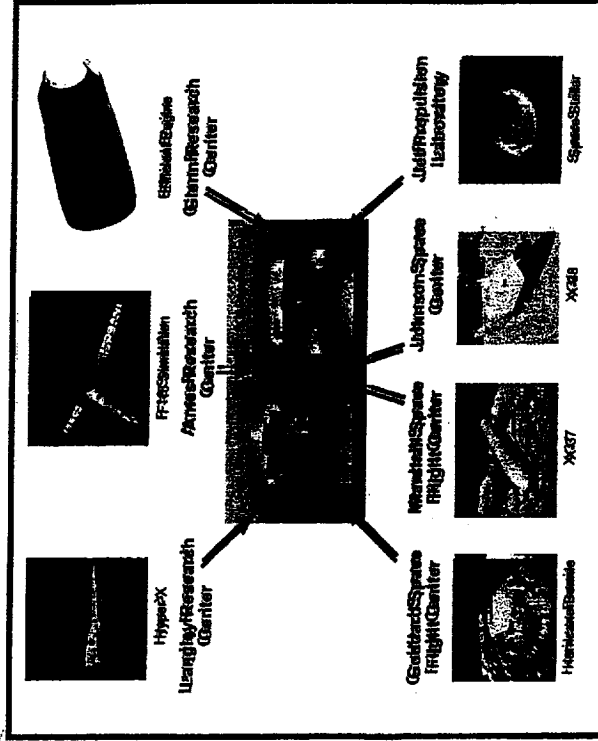


Ames Research Center | www.arc.nasa.gov

The Consolidated Supercomputing Management Office (COSMO) is a principal center responsible for the Ames Research Center. The primary goal of COSMO is to meet the high performance computing requirements for all enterprises, while realizing an optimal return investment through research and development management of NASA's high performance computing assets. It is responsible for the acquisition, maintenance, operation, management, upgrade, and cost effectiveness of the NASA's supercomputer resources, regardless of the platform. Operations and maintenance support are provided to NASA research and development programs and to ensure computing programs.

COSMO is responsible for establishing the High End Computing (HEC) goals for the agency, designing the strategy and action plan, and identifying cost-effective requirements. In addition, COSMO is responsible for establishing and maintaining uniform policies, reporting processes, providing resource usage and financial management, and ensuring that the NASA-wide approach is used in providing supercomputing services and in acquiring new supercomputing capabilities.

POC: Robert Schiner
<http://www.nasa.gov/COSMO>



Dave Gambrel
ADA 2.0000 2/10/00 01:00:00



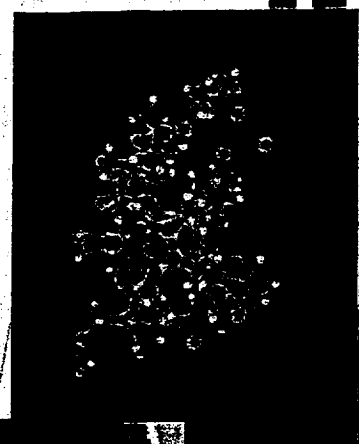
GROWLER: A COMPONENT-BASED FRAMEWORK FOR DISTRIBUTED/COLLABORATIVE SCIENTIFIC VISUALIZATION AND COMPUTATIONAL STEERING

Ames Research Center | www.arc.nasa.gov

growler provides an infrastructure that allows various kinds of functional modules to be "wired" together to create distributed and collaborative scientific applications. The individual components can be local or remote — that is, an application can be instantiated on a single machine, or it can be distributed over a network.

Scientific visualization applications are typically decomposed into viewer components and data components. The viewers perform graphics and take in user input. The data components provide data — either from archival databases, ongoing simulations, instruments, analysis procedures, or from some combination of these.

growler allows users to combine a diversity of viewer and data components to create a wide variety of custom-tailored utilities. For example, multiple viewers can be attached to a data stream to create a collaborative facility. Or, a single viewer can receive input from multiple data sources, creating a single "console" for overseeing control and results from a multi-stage distributed computation.



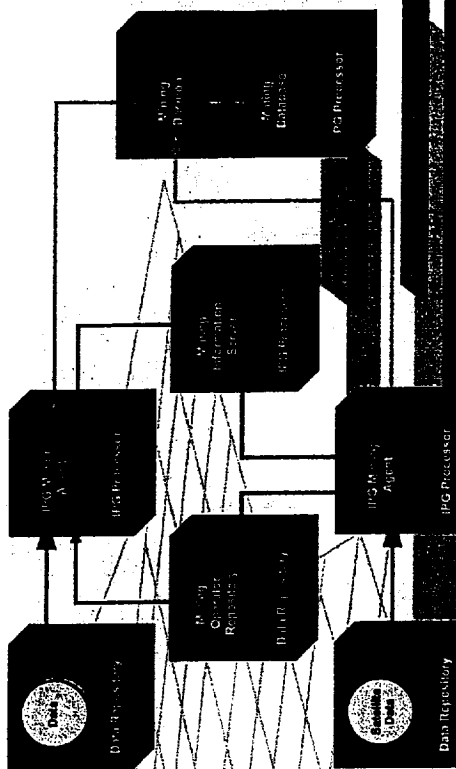
Bryan Green / Chris Henze

Ames Research Center | www.arc.nasa.gov

The Grid Miner uses NASA's Information Power Grid (IPG) to extract knowledge from remotely sensed satellite data using computationally intensive techniques.

The Grid Miner uses agents and the IPG to:

- Stage mining agent(s) and a mining plan to one or more IPG processors
- Consult the Mining Information Server to identify the Mining Operator Repository appropriate to the type of IPG processor to be used for mining
- Download needed mining operators from Operator Repository to support mining plan
- Acquire from the Mining Database the repository addresses of data to be mined
- Download data to be mined from IPG-accessible data repositories
- Mine the data by applying the mining operators specified in the mining plan to the data
- Produce one or more files with the mining results



Tom Hinkle



DEBUGGING ON THE INFORMATION POWER GRID (IPG)

Ames Research Center | www.arc.nasa.gov

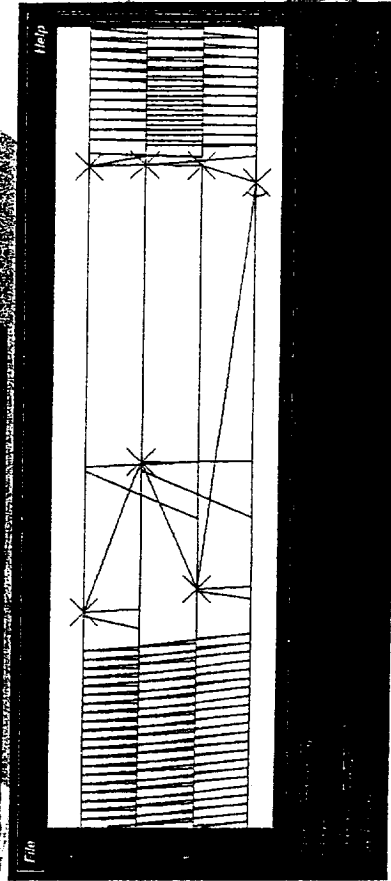
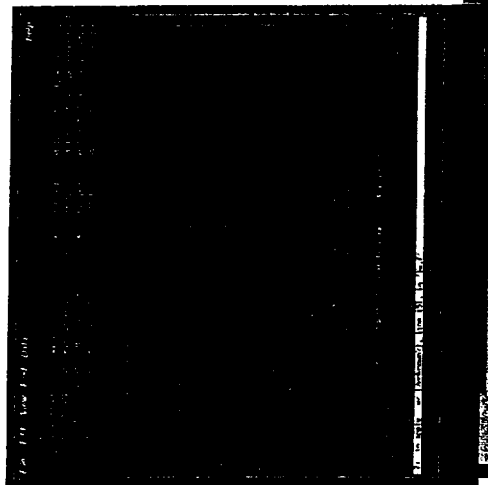
This demonstration shows three aspects of debugging on the Information Power Grid.

- We show how a debugger can control the execution of a computation running on a heterogeneous collection of machines. This is perhaps the key requirement of a debugger for computational grids such as the IPG. To our knowledge, no commercial debugger currently satisfies this requirement.
- We also show how a new approach that combines relative debugging and re-execution can automatically pinpoint errors in tool-parallelized codes with great accuracy. Through a combination of static analysis and re-execution, we can find the location where the parallel execution begins to differ from its sequential counterpart.
- Finally, we show how trace information collected during a computation can be used to provide a number of highly useful debugging operations. One of the key advances in this work is that the amount of trace information collected is greatly reduced. Some highly structured codes can be traced with fewer than 3 bits per traced event.

Effectiveness of trace compression

Trace source	Trace type	Size (MBs)		% compression
		before	after	
SP	p2d2	182.4	2.5	98.6
	ATMS	237.4	36.5	84.6
	PICL	381.8	65.5	82.8
LU	p2d2	724.7	4.6	99.3
	ATMS	647.4	120.7	81.3
	PICL	900.6	141.7	84.3

Another test: 50M events in 16.3MB < 3 bits/event !



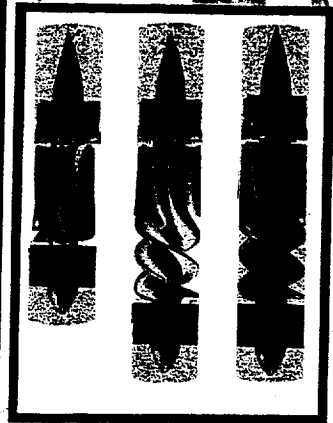
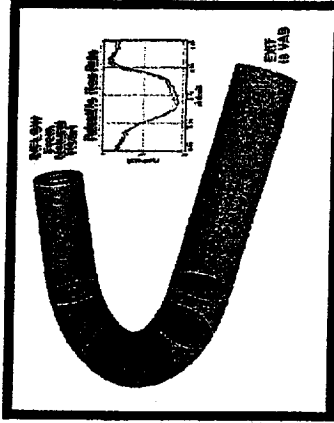
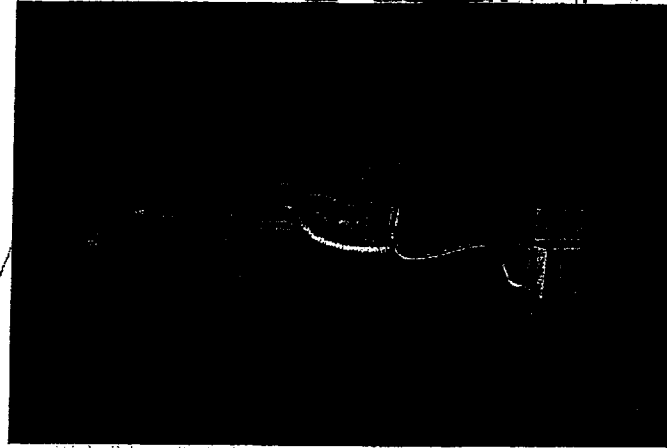
<http://www.nasa.gov/Tools/p2d2>

NASA DEBAKEY HEART ASSIST DEVICE

Approximately 20 million people worldwide suffer annually from heart failure, a quarter of them in America alone. In the United States, an alarmingly low 2,500 donor hearts are available each year.

The use of Computational Fluid Dynamics (CFD) technology has led to major design improvements on the heart assist device, enabling its human implantation. NASA Ames scientists employed NASA Shuttle main engine technology and NASA CFD modeling capabilities, coupled with the NASA Division's high performance computing technology, to make several design modifications that vastly improved the heart device's performance.

The research team investigated seven different designs, altering cavity shapes, blade curvature, inlet cannula shapes, and impeller tip clearance size. They then suggested three major design modifications to solve the problems of cell damage resulting from exposure to high shear stress and interrupted regions of blood flow in the DeBakey Ventricular Assist Device (VAD).



Ames Research Center | www.arc.nasa.gov

Cetin K. Kaya (1)

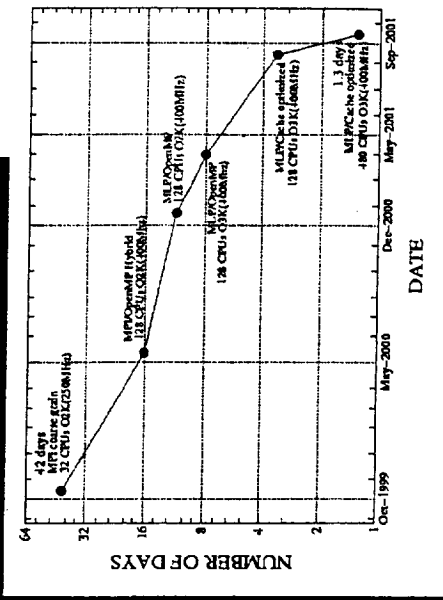


UNSTEADY TURBOPUMP FOR REUSABLE LAUNCH VEHICLE (RLV)

Ames Research Center | www.arc.nasa.gov

The objective of the current task is to provide a computational framework for design and analysis of the entire fuel supply system of a liquid rocket engine, including high-fidelity unsteady turbopump flow analysis. This capability is needed to support the design of pump subsystems for advanced space transportation vehicles that are likely to involve liquid propulsion systems. To date, computational tools for the design/analysis of turbopump flows are based on relatively lower fidelity methods. An unsteady, three-dimensional viscous flow analysis tool involving stationary and rotational components for the entire turbopump assembly has not been available for real-world engineering applications. The present effort will provide developers with information such as transient flow phenomena at start up, impact of non-uniform inflows, and system vibration and impact on the structure.

The research team used the 2nd Generation Reusable Launch Vehicle (RLV) baseline turbopump as a test case for evaluating the Multi Level Parallel (MLP) version of the incompressible flow solver, INS3D-code. The relative motion of the grid systems for the inlet vanes-impeller-diffuser interaction was obtained using overset grid techniques. Time-accuracy of the scheme has been evaluated with simple test cases. Unsteady computations for the 2nd Gen RLV baseline turbopump, which contains 114 zones with 34.5 million grid points, are carried out for four-impeller rotations on the SGI Origin-3000 systems at NASA Ames Research Center. One impeller rotation can be simulated in 3.5 days using 128 CPUs, or in 1.3 days using 480 CPUs. Particle traces and pressure surfaces are shown for these time-accurate simulations.





EXPLORATORY COMPUTING ENVIRONMENTS COMPONENT FRAMEWORK

We present a component framework designed to enhance the ability of NASA scientists to visualize, analyze and interact with huge data sets and associated distributed computations.

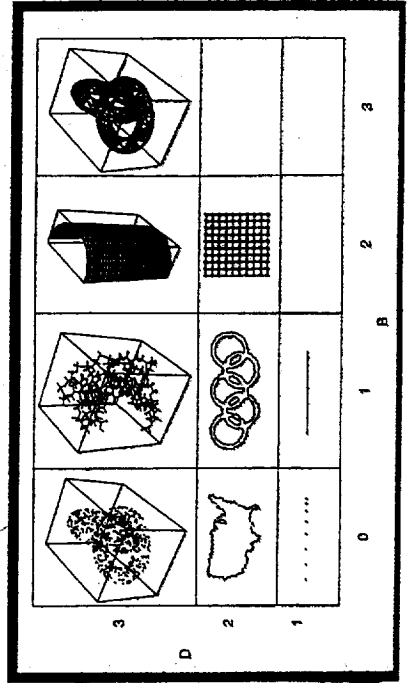
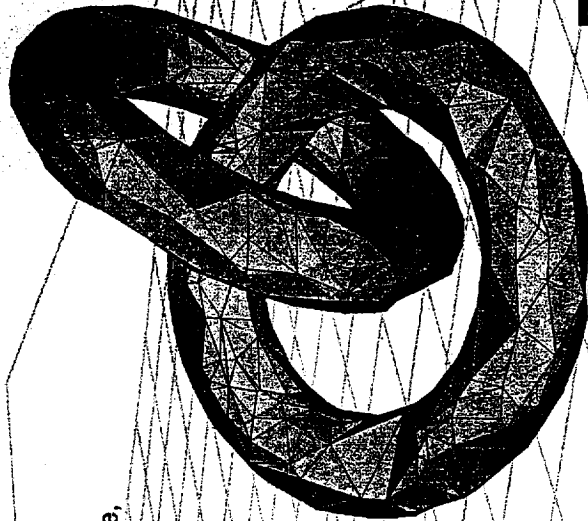
The components include:

- Data model: Field Model (FM)
- Metadata model: Active Metadata (AM)
- Visualization technique library (VisTech)
- Web technologies (Mars Landing Site PSE)
- Distributed object technology (growler)

Together, the components enable rapid prototyping and development of distributed computational environments. Application domains include aeronautics, earth science, biology, and nanotechnology.



Ames Research Center | www.arc.nasa.gov



Ames Research Center | www.arc.nasa.gov | Pat Moran



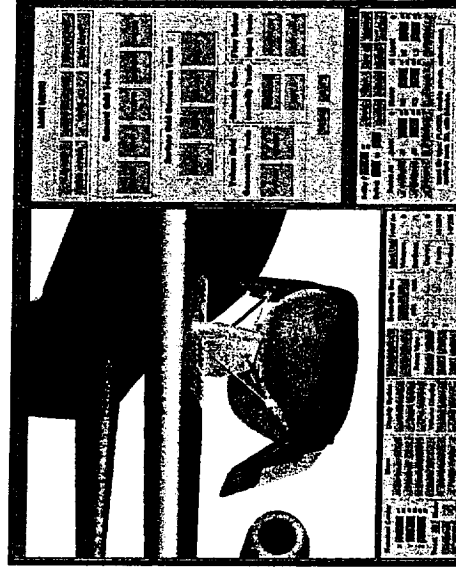
OVERSET Computational Fluid Dynamics Tools

Ames Research Center | www.arc.nasa.gov

The OVERSET grid approach is the most powerful and versatile method for computing viscous flow fields about complex geometries. Software developed at NASA Ames Research Center has been used by hundreds of analysts to compute flow over many types of commercial airplanes, military airplanes, helicopters, submarines, trucks, rockets, space vehicles, and to compute blood flow through a Ventricular Assist Device.

These software packages include: the Chimera Grid Tools, which contains many programs used for grid generation, pre- and post-processing; PEGASUS 5, which is used to join the overset meshes together; OVERFLOW, which solves the compressible Navier-Stokes equations; and INS3D, which solves the incompressible Navier-Stokes equations.

The first figure shows a screen image of the OVERGRID program, which is a GUI for the Chimera Grid Tools. The second figure illustrates a flow solution about the Space Shuttle Launch Vehicle, at an altitude of 33,000 feet, traveling at Mach 1.25. The color denotes the pressure coefficient on the vehicle surface, and the local Mach number on the symmetry plane. (Image courtesy of Reynaldo Gomez, NASA JSC).



Stuart Rogers

CONTROL AND OBSERVATION IN DISTRIBUTED ENVIRONMENTS

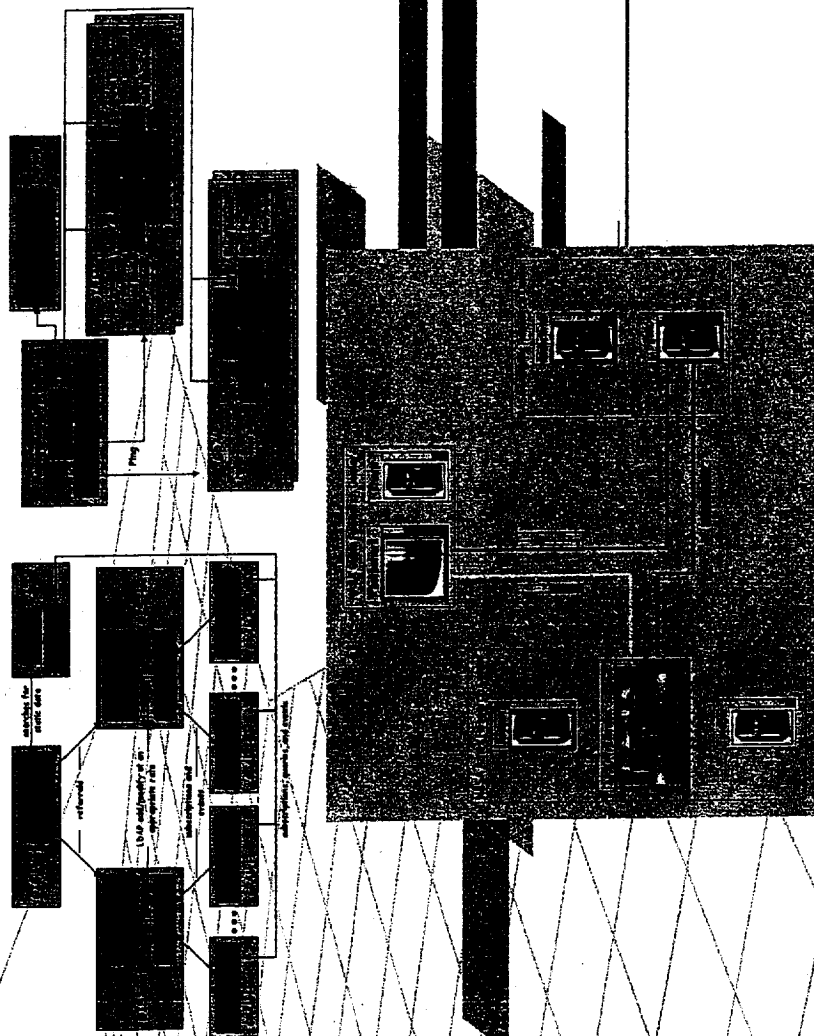


Ames Research Center | www.arc.nasa.gov

Researchers in the NAS division of NASA Ames Research Center have developed a software framework for control and observation in distributed environments, called CODE, for observing and controlling resources, services, and applications in large-scale distributed systems. CODE provides a secure, scalable, and extensible framework to make observations, reason about observations, and perform actions to adjust how an observed entity is operating.

This demonstration shows how an administrator can use CODE to manage a computational grid. Administrators can observe distributed resources such as computers, networks, and storage systems and determine how they are operating. Administrators can also observe and control grid services such as directory servers. Finally, administrators can use CODE to securely perform administrative tasks such as adding or removing grid users to distributed computer systems.

In ongoing work, we are using the CODE framework to provide grid information to a set of directory servers. Grid information stored in such a set of directory servers is typically called a Grid Information Service (GIS). A GIS maintains a somewhat up-to-date view of a computational grid and allows a user to search this information to locate computer systems of specific types, find resources that have appropriate loads and capacities, discover computer systems that have specific software packages, and so on.



Warren Smith



MLP SCALING ON NASA'S ORIGIN 1024 CPU SYSTEM

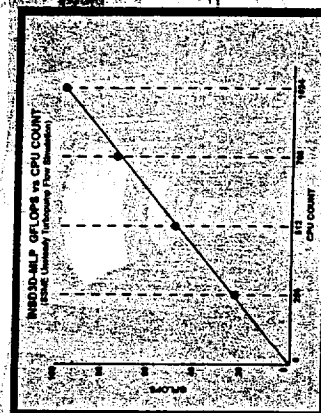
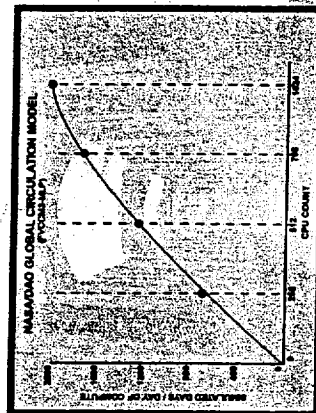
Ames Research Center | www.arc.nasa.gov

NASA Ames has developed a very simple and highly scalable method for parallelizing a number of production codes used in critical NASA missions. The new technique, called shared memory multi-level parallelism, or MLP, is vastly simpler than the classic MPI approach in use for years. It is a user-callable library of just 3 routines consisting of 150 lines of source code with latency characteristics about 10 times better than MPI on the same platform.

NASA applications in molecular modeling (COSMOS), computational fluid dynamics (OVERFLOW, INS3D), and data assimilation/climate modeling (DAS/FVCCM3) have been converted to the MLP approach with typical performance improvements of 10x (or more) over the previously available best results. Scaling to 1024 CPUs has been achieved in several cases.

These recent increases in code performance have had a major impact on NASA science. Highly resolved full aircraft simulations with the OVERFLOW code are now done in a few hours instead of weeks, allowing the aerospace industry to bring product to market years earlier. Climate predictions are 10 times faster than just a few months ago, allowing scientists to attempt simulations not possible just last year. Finally, simulations of turbomachinery for the Space Shuttle main engine are executing 10 times faster, allowing much more detailed and timely exploration of the behavior of new designs.

POC: James R. Taft
Director, Advanced Computing Technologies.



Jim Taft

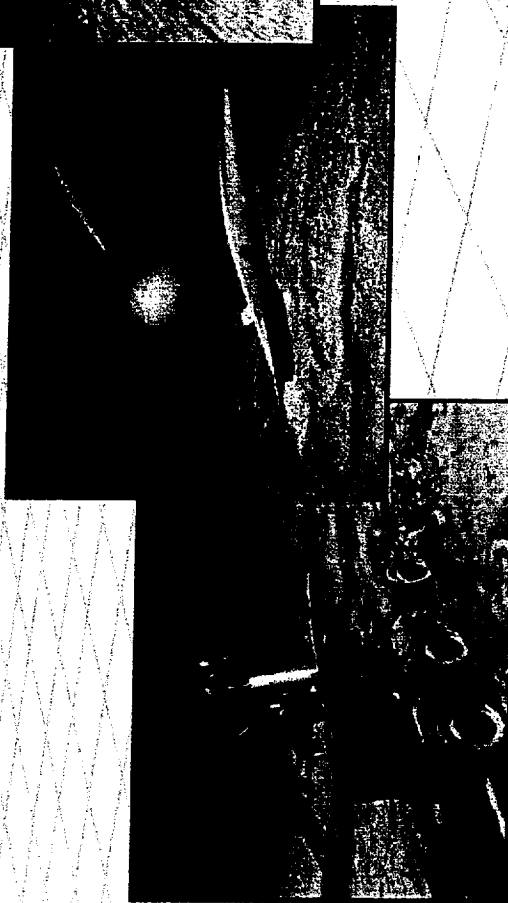
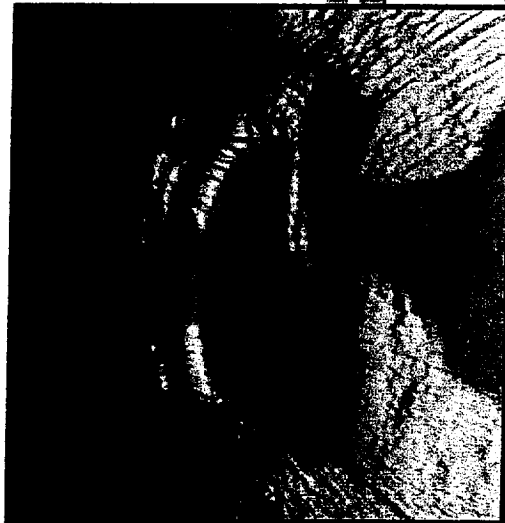
NASA

COMPUTING, INFORMATION, & COMMUNICATIONS TECHNOLOGY (CICT)

Ares Research Center | www.arc.nasa.gov

CICT GOAL

Enable NASA's scientific research, space exploration, and aerospace technology missions with greater mission assurance, for less cost, and with increased science return through the development and use of advanced computing, information and communications technologies.



Bill Vandelsem
NASA Research Center

NASA NAS GRID BENCHMARKS

Ames Research Center | www.arc.nasa.gov

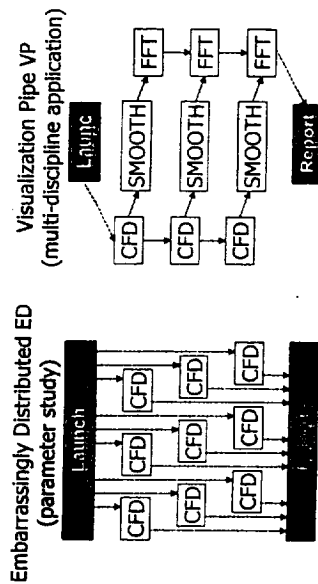
The NAS Grid Benchmarks (NGB) is a tool that tests the functionality and efficiency of grid environments such as NASA's Information Power Grid. NGB extends the methodology of the well-known NAS Parallel Benchmarks (NPB) to computational grids. It consists of four families of distributed applications defined as data flow graphs whose nodes are NPB codes. The arcs connecting nodes symbolize communication of whole field arrays between NPB codes.

In this demonstration we present the first complete "paper-and-pencil" specification of the NGB, which can be used by grid developers and operators to test their environments. We also show prototype implementations using Java and the Globus Metacomputing Toolkit. In its basic form, NGB measures turnaround time of its distributed applications. Our Globus implementation also reports queue time and CPU time. We show some different usage scenarios for the NGB that may give insight into the health and performance of grid environments.

- Goals:** Measure efficiency + functionality of modern Grid environment running (distributed) scientific application (includes public network performance)
- Audience:**
- Grid user: application engineer, researcher
 - Grid software/infrastructure developer
- Contents:** 4 compound tasks: 3 pseudo apps (LU, BT, SP) 2 kernels (MG, FT) from NAS Parallel Benchmarks
- Approach:** NGB1: Paper-and-pencil specification — 2001
NGB2: Source code implementations (NPB2.3 + Globus/Legion /CORBA/Grid Engine/ Condo/Java/Your_Grid_Environment/...)



NAS Grid Benchmarks NGB measure turnaround time of distributed applications



Rob Vander Wijnngaert



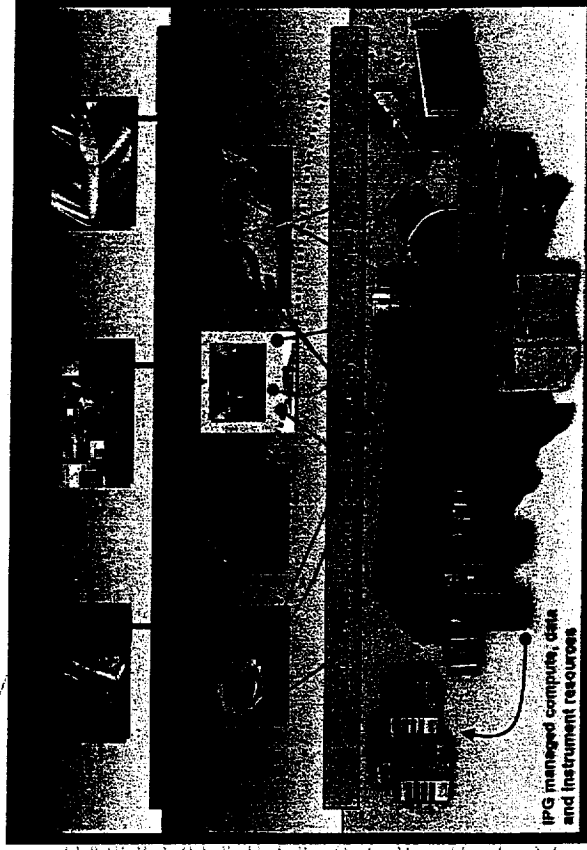
INFORMATION POWER GRID (IPG):

A LARGE-SCALE DISTRIBUTED COMPUTING AND DATA MANAGEMENT SYSTEM

Ames Research Center | www.arc.nasa.gov

Presentation Highlights:

- Information Power Grid (IPG)
 - description, motivation and goals
- NASA Applications driving IPG
 - Multidisciplinary simulations
 - Design process parameter studies
- Architecture
- IPG reference configuration
- IPG project milestones—accomplished
- Future plans



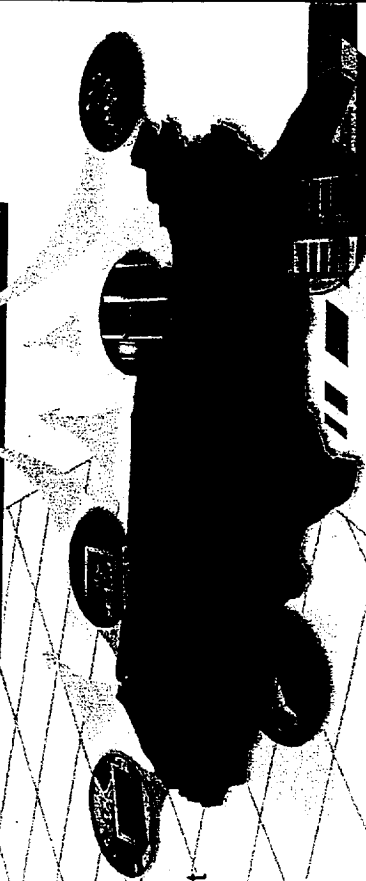
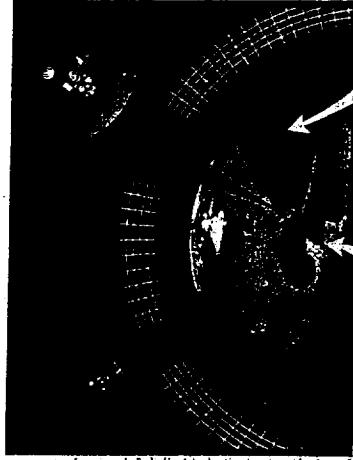
INFORMATION POWER GRID (IPG)

Ames Research Center | www.arc.nasa.gov

Grids are an emerging technology that provides seamless and uniform access to the geographically dispersed computational, data storage, networking, instruments and software resources needed for solving large-scale scientific and engineering problems.

NASA's Information Power Grid (IPG) project is developing and deploying such a computing and data grid, and its goal is to use NASA's remotely located computing and data system resources to build distributed systems that can address problems too large or complex for a single site.

The IPG provides a persistent infrastructure and will, on demand, locate and co-schedule the resources of multiple NASA centers needed to solve large-scale or widely distributed problems. IPG services will support workflow management frameworks that coordinate the processes inherent in distributed science and engineering problems of the NASA Enterprises: simulation, design, data collection, monitoring and control. IPG is a multicenter effort in collaboration with universities and other government agencies.



ARC-IPG-Program - 24W x 20h.ai



ILAB: PARAMETER STUDY CREATION AND SUBMISSION ON THE IPG

Ames Research Center | www.arc.nasa.gov

Parameter study creation and submission has been fully automated with this GUI-based tool.

In addition, the greatest impediment to user acceptance of the IPG/Globus model has been overcome: the job control language, which is required for submitting to the IPG/Globus processing environment, has been eliminated. Our GUI tool first automates the creation of parameter studies of arbitrary dimensions and then automatically creates all Globus job control language (RSL decks and auxiliary shell scripts) required to actually set up and invoke Globus jobs on the IPG.

We have demonstrated the ILab capabilities by creating a 12 by 16 case parameter study in Mach number and Alpha (angle of attack) for the X-38 Crew Return Vehicle. ILab automatically submitted these 192 flow solution cases to two separate Origin2000 parallel machines on the IPG testbed, using Globus protocols.

